Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Information Processing Letters 101 (2007) 126–128
Information Processing Letters
www.elsevier.com/locate/ipl

The Ford–Johnson algorithm still unbeaten for less than 47 elements ☆

Marcin Peczarski

Institute of Informatics, Warsaw University, ul. Banacha 2, 02-097 Warszawa, Poland

Received 11 May 2006; received in revised form 19 May 2006; accepted 4 September 2006

Available online 5 October 2006

Communicated by L.A. Hemaspaandra

Abstract

Using exhaustive computer search we show that sorting 15 elements requires 42 comparisons, and that for $n < 47$ there is no algorithm of the following form: "m and $n - m$ elements are sorted using the Ford–Johnson algorithm first, then the sorted sequences are merged", whose total number of used comparisons is smaller than the number of comparisons used by the Ford–Johnson algorithm to sort n elements directly.

Keywords: Analysis of algorithms; Sorting; Merging

We present recently obtained results in the computer search for optimal (i.e. minimum-comparison) sorting algorithms. Comparison counting is not the only measure of sorting performance. The number of memory accesses and cache hits may be of central importance in modern systems. However, as argued in [6, Ch. 5.3.1], research on minimum-comparison algorithms is a fascinating theoretical problem that "helps us to sharpen our wits for the more mundane problems".

Let $S(n)$ be the minimum number of comparisons sufficient to sort n elements. Let $F(n)$ be the number of comparisons required by the Ford–Johnson algorithm (FJA) [2] to sort n elements. It is known that $F(n)$ achieves the information-theoretic lower bound $C(n) \stackrel{\text{def}}{=} \lceil \log_2 n! \rceil$ for $n \leqslant 11$ and $n = 20, 21$. Hence the FJA is optimal in these cases, i.e. $S(n) = F(n) = C(n)$. For $n = 12, 13, \ldots, 19, 22$ we have $F(n) = C(n) + 1$. Performing an exhaustive computer search, Wells discovered that $S(12) = F(12) = 30$ [15,16]. Extending Wells's method, we showed that $S(13) = F(13) = 34$ [10]. Further development of this method allowed us to show that $S(14) = F(14) = 38$ and $S(22) = F(22) = 71$ [11]. In this paper we present results of two more computer experiments. The first one proves the following theorem.

Theorem 1. *Sorting* 15 *elements requires* 42 *comparisons.*

It turns out that the FJA is optimal for $n \leqslant 15$ and $n = 20, 21, 22$ elements. On the other hand, $n = 47$ is the smallest number of elements for which we know an algorithm better than the FJA. The algorithm was found by Schulte Mönting [13]. First, 5 and 42 elements

☆ This paper contains results obtained using computer resources of the Interdisciplinary Centre for Mathematical and Computational Modelling Warsaw University.

E-mail address: marpe@mimuw.edu.pl (M. Peczarski).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2006.09.001

are sorted by the FJA using 7 and 171 comparisons, respectively. Next, the sorted sequences are merged using further 22 comparisons. To sum up, this algorithm requires only 200 comparisons while the FJA requires 201 comparisons. The FJA is not optimal for infinitely many values of n, and all better algorithms we know are combinations of sorting and merging [7,8]. The other presented experiment proves the second theorem.

Theorem 2. *For $n < 47$ there is no algorithm of the following form: "m and $n - m$ elements are sorted using the FJA first, then the sorted sequences are merged", whose total number of used comparisons is smaller than the number of comparisons used by the FJA to sort n elements directly.*

To perform the first experiment we used the method described in [11]. We obtained that $C(15) = 41$ comparisons do not suffice to sort 15 elements and hence $S(15) = F(15) = 42$. The computation was distributed to a cluster of computers. About $78 \cdot 10^9$ posets were generated and stored on a disk. The whole computation took 17 554 hours of CPU time. Up to 24 processors (64-bit, 2 GHz) were used in parallel. More detailed description can be found in [12].

Let $M(m, k)$ be the minimum number of comparisons sufficient to merge m and k elements. To prove Theorem 2, we have to check that for $n < 47$ and $m \leqslant n/2$ the following inequality holds:

$$F(m) + F(n - m) + M(m, n - m) \geqslant F(n).$$

Table 1
The values $M(m, n - m)$

n	m																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	1																
3	2																
4	2	3															
5	3	4															
6	3	5	5														
7	3	5	6														
8	3	6	7	7													
9	4	6	7	8													
10	4	6	8	9	9												
11	4	7	8	10	10												
12	4	7	9	10	11	11											
13	4	7	9	11	12	12											
14	4	7	10	11	12	13	13										
15	4	8	10	12	**13**	14	14										
16	4	8	10	12	13	**15**	15	15									
17	5	8	11	13	14	15	16	16									
18	5	8	11	13	14	16	17	17	17								
19	5	8	11	13	15	16	17	18	18								
20	5	8	11	14	15	17	18	19	19	19							
21	5	9	12	14	16	17	19	20	20	20							
22	5	9	12	14	16	18	19	20	21	21	21						
23	5	**9**	12	15	17	18	20	21	22	22	22						
24	5	9	**12**	15	17	19	20	22	⩾ 22	23	23	23					
25	5	9	12	**15**	**17**	19	21	⩾ 22	⩾ 23	24	24	24					
26	5	9	13	**15**	**18**	**20**	21	⩾ 22	⩾ 23	25	25	25	25				
27	5	9	13	16	**18**	**20**	**22**	⩾ 23	⩾ 24	⩾ 25	26	26	26				
28	5	9	13	16	**18**	**21**	**⩾ 22**	**⩾ 23**	**⩾ 24**	**⩾ 25**	27	27	27	27			
29	5	10	13	16	**19**	**⩾ 21**	**⩾ 22**	**⩾ 24**	**⩾ 25**	**⩾ 26**	⩾ 27	28	28	28			
30	5	10	13	16	19		**⩾ 23**	**⩾ 24**	**⩾ 26**	**⩾ 27**	**⩾ 27**	29	29	29	29		
31	5	10	13	17	19			**⩾ 24**	**⩾ 26**	**⩾ 28**	**⩾ 28**	30	30	30	30		
32	5	10	14	17	19				**⩾ 26**	**⩾ 28**	**⩾ 29**	**⩾ 30**	31	31	31	31	
33	6	10	14	17	20					**⩾ 28**	**⩾ 29**	**⩾ 30**	32	32	32	32	
34	6	10	14	17	20					**⩾ 28**				33	33	33	33
44	6	**11**	15	19	22												
45	6	**11**	**15**	19	22												
46	6	11	**15**	**19**	**22**												
47	6	11	**15**	**19**	**22**	**⩾ 25**	**⩾ 27**										

The information-theoretic lower bound tells us that

$$M(m, n-m) \geqslant \left\lceil \log_2 \binom{n}{m} \right\rceil.$$

Therefore we need to check only those combinations of m and n for which

$$F(m) + F(n-m) + \left\lceil \log_2 \binom{n}{m} \right\rceil < F(n). \qquad (1)$$

The formula for $F(n)$ is well known, see e.g. [6]:

$$\begin{aligned} F(n) &= \sum_{k=1}^{n} \left\lceil \log_2 \frac{3}{4}k \right\rceil \\ &= nd - \left\lfloor 2^{\lfloor \log_2(6n) \rfloor}/3 \right\rfloor + \left\lfloor \frac{1}{2}\log_2(6n) \right\rfloor \\ &= \left(n + \frac{1}{2}\right)d - \frac{4}{3}2^d + \frac{1}{12}(-1)^d + \frac{5}{4}, \end{aligned}$$

where $d = \left\lceil \log_2 \frac{3}{4}n \right\rceil$.

Given this exact formula for $F(n)$, we only need to worry about the values of $M(m, n-m)$ for which the information-theoretic lower bound is not sufficient to prove Theorem 2. Table 1 is what we know about $M(m, n-m)$, via a combination of formulas and computer programs. Exact formulas are known only for $m \leqslant 4$ [3–6,9,13] and $\frac{2}{5}(n-1) \leqslant m \leqslant \frac{1}{2}n$ [6,14]. All values $M(m, n-m)$ for $m \leqslant 10$ and $n-m \leqslant 10$ can be found in [6]. For others, we have used computer programs similar to the ones used in [11]. Using computer programs is acceptable due to the sheer non-trivial nature of computing $M(m, n-m)$ for all cases: see [13] for details of this complexity. Some values were computed using both methods to check correctness of the implementation. All combinations of $n \leqslant 47$ and $m \leqslant n/2$, which (1) holds for, are marked in Table 1 by setting appropriate value $M(m, n-m)$ in bold. Because computation time grows in the number of considered comparisons, only the lower bound was computed for some values. These values are preceded with the sign $\geqslant$. Formulas (5.3.2.10)–(5.3.2.14) in [6] helped us to obtain some of these bounds. It should be mentioned that all presented lower bounds are bigger than the information-theoretic lower bound. The obtained results confirm Theorem 2.

A computation was also performed for $n = 47$ to verify the result from [13] that the FJA is not optimal in this case. This enhances our conviction that carried out computations are correct.

The algorithm used in the second experiment is basically the same as in [11]. There are just two changes. First, we start analysis from the set containing one poset which consists of two chains with m and $n-m$ elements, respectively. Second, we consider only posets of width two, and hence the polynomial algorithm for counting linear extensions published in [1] can be applied. The experiment was performed on a cluster of computers and took about 32 000 hours of CPU time. However most of this time, i.e. about 29 000 hours, was spent on the computation that for $n = 31$ we have $M(10, 21) \geqslant 28$. More detailed description can be found in [12].

In summary, we hope that we have ascertained some interesting facts about minimum-comparison sorting. The author wishes to thank referees for helpful comments.

References

[1] M.D. Atkinson, H.W. Chang, Computing the number of mergings with constraints, Information Processing Letters 24 (1987) 289–292.

[2] L. Ford, S. Johnson, A tournament problem, American Mathematical Monthly 66 (1959) 387–389.

[3] R.L. Graham, On sorting by comparisons, in: A.O. Atkin, B.J. Birch (Eds.), Computers in Number Theory, Academic Press, London, 1971, pp. 263–269.

[4] F.K. Hwang, Optimal merging of 3 elements with n elements, Memorandum, Bell Telephone Laboratories, Murray Hill, NJ, 1977.

[5] F.K. Hwang, S. Lin, Optimal merging of 2 elements with n elements, Acta Informatica 1 (1971) 145–158.

[6] D.E. Knuth, Sorting and Searching, second ed., The Art of Computer Programming, vol. 3, Addison–Wesley, Reading, MA, 1998.

[7] G.K. Manacher, The Ford–Johnson algorithm is not optimal, Journal of the Association for Computing Machinery 26 (1979) 441–456.

[8] G.K. Manacher, T.D. Bui, T. Mai, Optimum combinations of sorting and merging, Journal of the Association for Computing Machinery 36 (1989) 290–334.

[9] Y. Nissenbaum, On merging N ordered elements with three elements, SIGACT News 9 (4) (1978) 14–16.

[10] M. Peczarski, Sorting 13 elements requires 34 comparisons, in: Proceedings of the 10th Annual European Symposium on Algorithms, in: R. Möhring, R. Raman (Eds.), Lecture Notes in Computer Science, vol. 2461, Springer, Berlin, 2002, pp. 785–794.

[11] M. Peczarski, New results in minimum-comparison sorting, Algorithmica 40 (2004) 133–145.

[12] M. Peczarski, Komputerowo wspomagane badanie zbiorów częściowo uporządkowanych (Computer assisted research of posets), PhD thesis, Warsaw University, Warsaw, 2006.

[13] J. Schulte Mönting, Merging of 4 or 5 elements with n elements, Theoretical Computer Science 14 (1981) 19–37.

[14] P. Stockmeyer, F. Yao, On the optimality of linear merge, SIAM Journal on Computing 9 (1980) 85–90.

[15] M. Wells, Applications of a language for computing in combinatorics, in: Information Processing 65 (Proceedings of the 1965 IFIP Congress), North-Holland, Amsterdam, 1966, pp. 497–498.

[16] M. Wells, Elements of Combinatorial Computing, Pergamon Press, Oxford, 1971.